Spam Filtering using Local-global Bayesian Classifier

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Software Engineering

Submitted By

Rohit Kumar Solanki

Roll No. 801331023

Under the supervision of:

Mr. Karun Verma
Assistant Professor
CSE Department

Mr. Ravinder Kumar
Assistant Professor
CSE Department

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Spam Filtering using Local-global Bayesian Classifier*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mr. Karun Verma and Mr. Ravinder Kumar* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Rohit Kumar Solanki)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Mr. Karun Verma)
Assistant Professor
CSE Department

(Mr. Ravinder Kumar)
Assistant Professor
CSE Department

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

i

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life.

This work would not have been possible without the encouragement and able guidance of my supervisor's **Mr. Karun Verma** and **Mr. Ravinder Kumar**. I thank my supervisor's for their time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable.

I am equally grateful to **Dr. Deepak Garg**, Associate Professor and Head, Computer Science & Engineering Department, a nice person, an excellent teacher and a well – credited researcher, who always encouraged me to keep going with work and always advised me with his invaluable suggestions.

I will be failing in my duty if I don't express my gratitude to **Dr. S.S. Bhatia,** Senior Professor and Dean of Academic Affairs, Thapar University, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable.

Last but not least, I would like to thank my family whom I dearly miss and without whose blessings none of this would have been possible. To my parents, I own thanks for their wonderful love and encouragement. I would also like to thank my brother, since he insisted that I should do so. I would also like to thank my close friends for their constant support.

Date: July, 2015
Place: Thapar University, Patiala

(Rohit Kumar Solanki)

ii

ABSTRACT

Spam is an email, which is usually sent in bulk by the sender. Unlike legitimate mails, there is no agreement between the receiver and the sender of the mail. That's why they are also termed as unsolicited mails. To prevent the delivery of this so called spam messages, an automated tool called a spam filter is used to recognize spam. As there is no single definition of spam, it is difficult to formulate rules to block such unwanted messages. There are several techniques used to stop those unwanted messages. It is not full proof against spam, even with the introduction of new state of the art techniques. Some of the techniques are based on manually configured rules, others rely on statistical calculations for adapting themselves according to the current situation.

In this thesis, a novel learning framework for classification of messages into spam and legit is proposed. Naive Bayes (NB) model is a statistical filtering process which uses previously gathered knowledge. Instead of using a single classifier, the use of local and global classifier, based on the Bayesian hierarchal framework is proposed. This helps in achieving multi-task learning, as simultaneous extraction of knowledge can be achieved while achieving classification accuracy. Knowledge among different task can be shared while learning for task specific.

Table of Content

List of Figures

Figure No. Figure Name **Page No.**

List of Tables

Table No. **Table Name** **Page No.**

Chapter 1

INTRODUCTION

1.1 Introduction of Spam Filter:

It has been estimated that more than 70% of the emails are unwanted messages or spam [3]. This increase in the number of spam messages has resulted in the decrease in productivity. There is no precise definition of spam message. It can be an unwanted advertisement or it can be a message written in unfamiliar languages. Furthermore, these messages are defined as Unsolicited Commercial Email (UCE) or Unsolicited Bulk Email (UBE). The vast amount of spam being sent by the spammer wastes resources on the internet, wastes the time of the users and also may be unsuitable for the children's due to their content. During the early period, most of the spam filters were instances of knowledge engineering using handcrafted rules. But due to common, and also most of the times, publicly accessible rules, it was easy for the spammers to bypass filters. So new types of filters were introduced with the focus on machine learning for the automatic rules creation. Spam filters differ from others Text categorization tasks [27]. It is not the content, but the unsolicited nature definitively the spam. Similarly the class of legitimate message may also span a number of diverse subjects [18]. Also, there is a penalty involved in classifying a legitimate message as spam, which ultimately helps in improving the rules. In figure 1, general working of spam filter is shown.

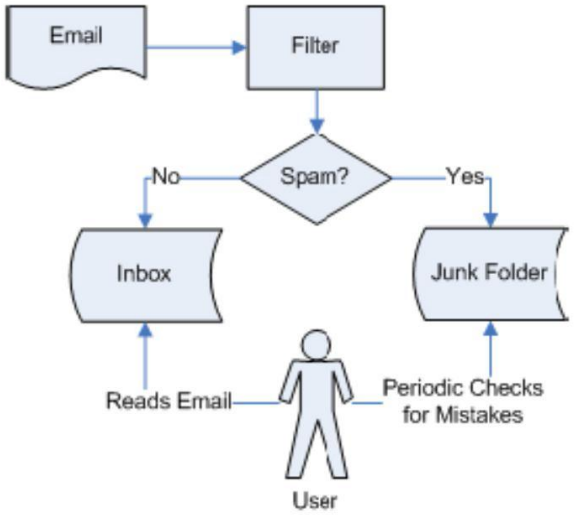


Figure 1 General Working of Spam Filter [14]

1.2 Classification of Spam filter Algorithms:

Spam filtering techniques can be mainly divided into four broad categories as shown in figure 2. Each one has a different method of handling spam messages.

1.2.1 List Based Filtering

These types of filters, treats incoming message according to the previously learned knowledge of the sender, and categorize mail accordingly [8].

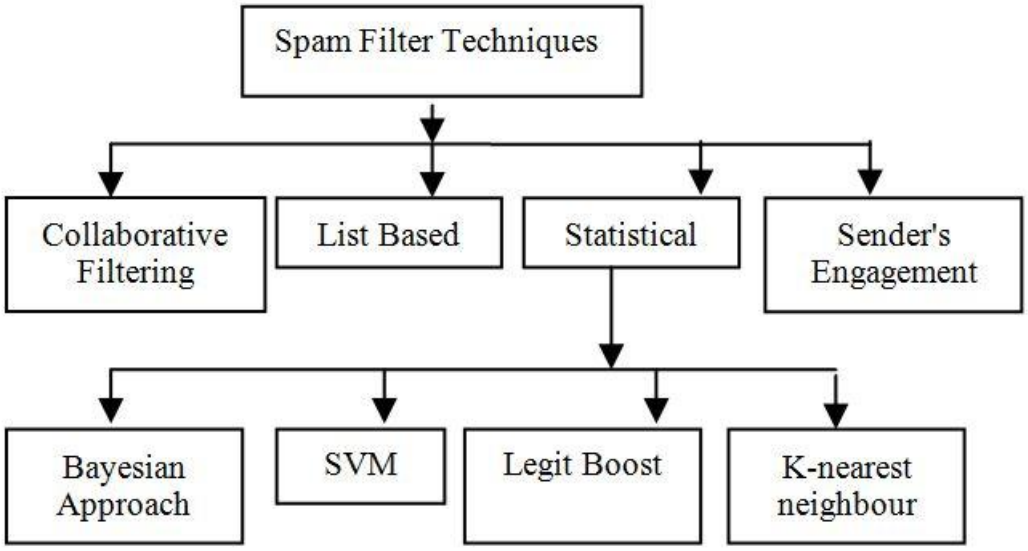


Figure 2 Different Spam Filtering Techniques

1.2.1.1 White List

The white list method works on principle of checking the sender in the trusted-user list. It works exactly opposite to the black list filtration. Any sender, who is present in the database is allowed to send mail. It is a strict filter which allows only those senders present in the list, thus leaving out those who doesn't have the appropriate privilege.

1.2.1.2 Black List

Black list are the records of the senders flagged as a spammer. This record contains IP addresses or email addresses used by the sender. When an incoming message arrives, it is checked with the entries in the list. Another version of this method is to use a third party black list for filtering. On the arrival of the incoming message, receivers

filter contact with the third party system and compare message to the newly updated black list. This method is known as a Real-Time Blackhole List.

1.2.1.3 Grey List

It is based on the assumption that most of the spammers will try to send bulk emails one time. So the model rejects the email first time and sends a failure receipt to the sender. If the sender is a genuine user, then for second time delivery, it is allowed by the filter and the user is added to the legitimate user database. But this method has a drawback as time sensitive mails cannot be rejected 1st time. This delay in the delivery is an inherent problem of this model.

1.2.2 Collaborative Filtering

Collaborative Filtering System is based on users-feedback approach. It collects information from millions of users around the globe. This approach uses a flagging mechanism, done by the users. Upon reaching a certain threshold flags, mail is reported to the central database as a genuine message or a spam. And if a message is marked as a spam by the model, it's automatically stopped by the system from reaching other users.

One of the main advantage of this model is that, for a large active user base, this model can quickly stop a spam outbreak, sometimes within a matter of minutes. But if a large number of spammers pretend to be legitimate users of the system, they can pass spam messages by labeling them as genuine message.

1.2.3 Content Based Filtering:

This method mainly focuses on the content of the method rather than list of trusted users, or feedback from different users using the filter. Instead of following a strict rule based on the list, this technique uses words or phrase as features to determine the message as the genuine message or spam.

There are many types of content based filters such as word based filtering, which solely focus on the words present in the message. If the message contains certain words which are generally found in the spam, then filter automatically classifies the message as spam.

Another type of filters, known as Heuristic filters, which determines the class of the message according to the words contained in the message. Each word is assigned some points such as words like 'Lottery' receives higher points as they are generally present in the spam, while terms like ‘SBI’ has low points. The filter than add up points given to each word and determine the overall score. If the result is higher than a threshold value than message is termed as spam. Since both the method calls for manual configuration of the rules, they don't provide the flexibility and doesn't adapt according to the alteration in the incoming flow. This problem was overcome by the introduction of the statistical spam filters. It uses mathematical models to predict the legitimacy of the incoming mail, according to the knowledge gathered by the classifier from the previous data.

1.3 Different types of Classifier

1.3.1 Single Classifier Techniques:

Using statistical model to classify text based on trained classifier is very efficient techniques. Earlier spam filter models were based on single classifier usage. Some of the techniques used in the spam detection are described briefly. Detailed analysis of these methods is shown in the survey. Naive Bayes classifier is based on the Bayes theory of probability [1][2][23]. It uses the assumption that all the features are mutually independent as shown in figure 3. It calculates the posterior probability of the text based on the probability of the feature space and class.

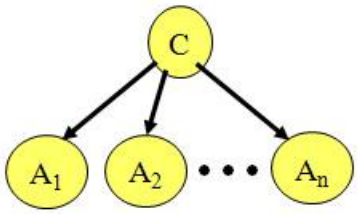


Figure 3 Naive Bayes Models

A supervised learning [3] approach to classification between spam and non-spam messages is implemented in the Bayesian Filter. It uses Bayesian classifier and learns the difference between spam and non-spam through a dataset and then applies that knowledge to binary classification decision whenever a new mail is presented. It uses bags of words features to identify spam email, an approach which is mostly used in

text classification. For example, a word “news” often appears in legitimate mail then the probability of “news” indicating spam is near to zero. When a mail arrives, Bayesian classifier is used to calculate the probability of mail being spam. This type of filter can adapt automatically, i.e. the message can be used to train the filter further.

Apart from Bayesian technique, there are other techniques such as AdaBoost classifier, KNN classifier [25], Fisher Robinson inverse Chi-Square function. Chi-Square was proposed by Gary Robinson which uses a probability function named as “Robinson’s degree of belief”. There is another function proposed by Sir Ronald is Fisher- Robinson Inverse Chi-Square Function.

AdaBoost classifier investigates the performance of active learning using confidence based sampling using Boosting. Another task is to train a classifier which helps in obtaining scoring function which can be further used in classification of mail as spam or non-spam. AdaBoost (Adaptive Boosting) is an iterative algorithm which repeatedly updates the weight of wrongly and correctly classified records. While the weight of wrongly classified record is increased, the weight of correctly classified record is decreased. Making the new classifier more sensitive to wrongly classified records. Initially, classifier is manually trained by the user.

KNN (K Nearest Neighbor) classifies the mail based on their similarity to the class of mails [31]. The k – means is basically used to partition the data in such a way that while the inter cluster similarities is low, intra cluster similarities are high. KNN classifier uses Euclidean distance to find k nearest neighbors.

Support Vector Machines are based on the concept of decision planes that define decision boundaries [21]. A decision plane is one that separates between a set of objects having different class memberships, SVM finds an optimal hyper plane with the maximal margin to separate two classes, which requires solving the following optimization problem. Apart from these techniques, there are some other methods such as TF-IDF, Decision tree, Random forest, Neural Network etc.

A single classifier acts as a centralized system, treating every user as equal. It is easy to train using labelled data. The main disadvantage of single classifier filter is the inherent disadvantages of algorithms using in the classifier. Using multiple classifiers can overcome such inherent problems.

1.3.2 Multiple Classifiers Techniques:

In recent times, a combination of different classifiers for the spam detection has been proposed. As different classifiers show different accuracy, performance and speed, it can be beneficial to combine classifiers for improvement. Research has shown many combinations of algorithms, and application of practical problems have shown the advantage of the ensemble over individual algorithm. The combination of the classifiers can be parallel, which is generally to improve the accuracy of the proposed model, or sequential, which is mainly used for the acceleration of classification of a large category set.

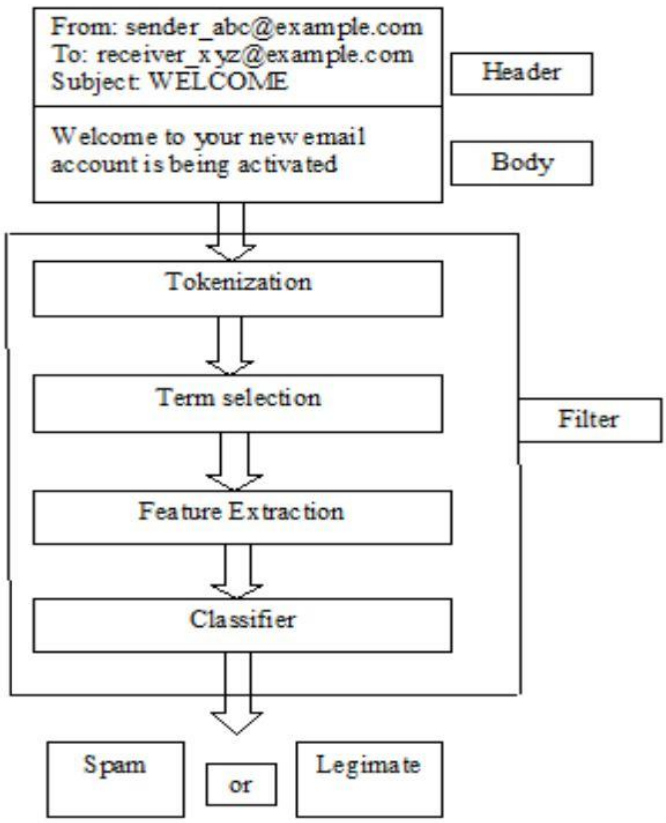


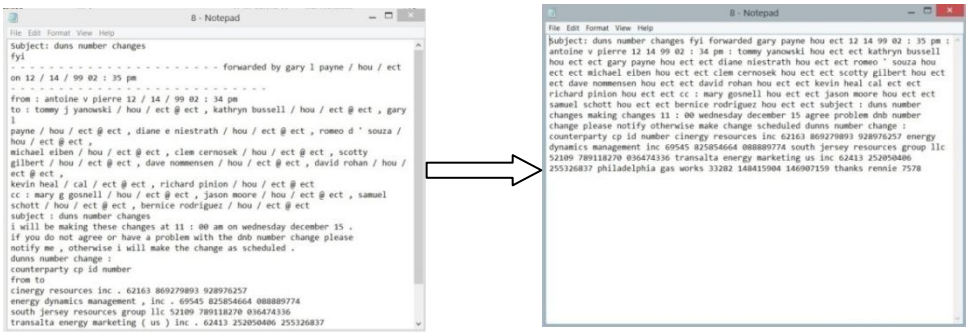
Figure 4 Different phases of Spam Filter

1.4 Steps involved in Naive Bayes Classification:

The objective of this project is to implement a Bayesian Spam Filter to differentiate between spams and also to measure its efficiency using various cost effective measures. The Bayesian technique is one of the fundamental DM technique. It is based on the probability theory. Following are the steps, as shown in figure 4, involved in the classification of spam using Bayesian model:

- **Pre-processing**:

  Higher percentage of stop words are present in emails. Inclusion of these words decreases the overall performance of spam filters as they are generally part of both type of mails, and are an essential part of grammar. Removal of these words and selecting those features which are part of the summary of a message improves the performance of the classifier. In this phase, prepossessing of all incoming messages into a defined format, and removal of unwanted words, is done in order to make messages more feasible to be handled by model. Removal of all the irrelevant data such as HTML tags, etc. and selecting all the suitable segments for processing (e.g. Headers, body, etc.).



(a) Normal Text **(b) Processed Text**

Figure 5 Pre-processing of mail and removing stop words

- **Tokenization**:

  Tokenization is the process of dividing incoming mail in the form of tokens. Some classifiers such as Naive Bayes, K-NN etc. represents tokens as strings or words. All the relevant data is divided in the form of tokens

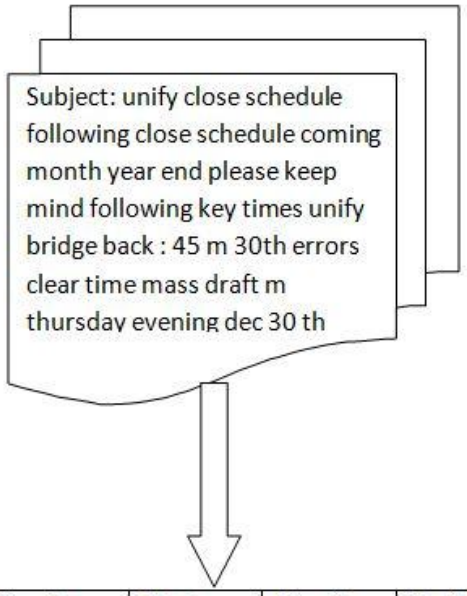
also known as semantically coherent segments (for e.g. sentences divided into words).

- **Representation**:

  Further conversion is done by representation. It is the conversion of messages into an attribute value par vector, where attribute are nothing else but previously defined tokens with their values as binary, frequency etc. Deletion of less predictive attributes is done in the selection process.

- **Learning:**

  In this step, labeled data (email) is used to learn classifier. Each labeled data is divided in the form of tokens, attributes are assigned, feature vector are extracted, and tokens are placed in the conditional probability table. During the learning phase, conditional probabilities for all the tokens is calculated. After that posterior probability with respect to tokens is summarized.



Token Name	Good Count	Bad Count	Good Prob	Bad Prob	Spam Prob
Subject	0	0	0	0	0
unify	0	0	0	0	0
close	0	0	0	0	0
schedule	0	0	0	0	0

Figure 6 Feature Extraction and Attribute's assignment

- **Performance Evaluation**:

  Performance evaluation of a spam filter model is done through calculating the accuracy of classifying mails in the correct category. For calculating the performance of the model, following parameters are calculated.

  - *Spam Recall*:

    Spam Recall is the ratio of number of spam messages classified as spam to total number of spam messages.

  - *Spam Precision:*

    In simple term, Spam Precision is the fraction of result classified as positive, which are indeed positive. It means the ratio of number of spam classified as spam to number of all the messages that are classified as spam.

  - *Error Rate*:

    It defines the ratio of misclassification of both spam and ham to total number of messages.

  - *Accuracy*:

    Accuracy is the exact opposite of Error Rate.

Chapter 2

LITERATURE SURVEY

Statistical Spam filtration models outperforms other models by a large margin. They are flexible in nature, can adapt according to the users, platform, and incoming stream. The accuracy rate is very high for such models. In this survey, the basic idea is to study different spam filtration techniques using mathematical foundation. This survey contains details of both single as well as hybrid classifiers. More emphasis is to study and analyse different techniques, and improve the accuracy of the Naive Bayes classifier using hybrid local-global classifier, a two stage classifier approach. After detailed survey, during which numerous different techniques are studied, proposed work theme is discovered, which includes classifier improvement, knowledge sharing between different users and the implementation to improve overall accuracy of the model.

2.1 Work related to Pre-processing

2.1.1 Stop Word

For Information Retrieval and text mining, stop word removal is an important step. Words which have the most frequency in the message, doesn't convey any message for the filter. I English dictionary, they are the pronouns, preposition, conjunction etc. There are more than 400 stop words present in the English language, but there is no single universal list of stop words. Examples of such words include 'the', 'of', 'and', 'to'. The first step during pre-processing is to remove these Stop words, which has proven as very important [30]. Also, most of the present work uses the SMART stop word list suggested in the paper.

2.1.2 Stemming

In proposed model, truncating stemming algorithm is required and thus survey contains a brief study of the following method. Stemming is a pre-processing step in this model [7]. It is one of the important tasks of the Information Retrieval system. The main task of the stemming is to reduce the word (stem the words) into its root form. Words can be a noun, adjective, verb, adverb, etc. Thus, stemming algorithms reduces the derivationally related form of a word to its base form. It also deduces inflectional form. For example, the words teacher, teaches, teaching, teach all can be

stemmed to the word 'teach'. In the present work, the Porter Stemmer algorithm [20, 21], which is the most commonly used algorithm in English, is used.

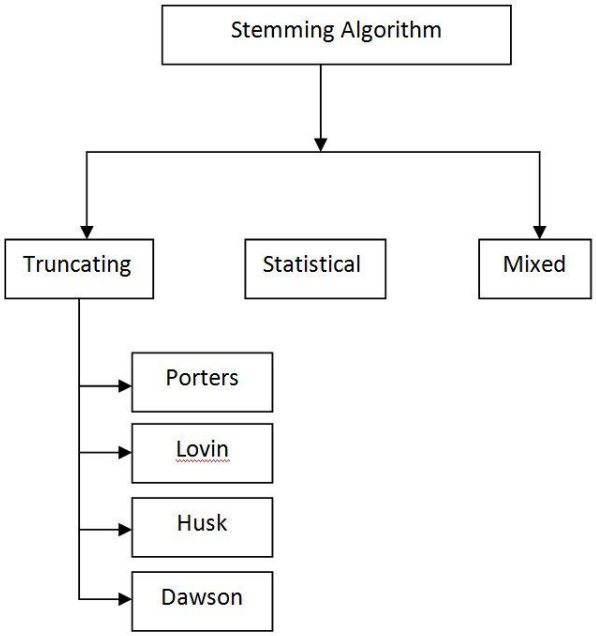


Figure 7 Stemming Algorithms

2.2 Work related to Classifier

Spam filter model uses the binary classification of text messages. It is a type of machine learning model, which can be trained by using labelled data, to classify incoming messages. Earlier, spam filters were based on single classifier. Some of the algorithms used for single classifier spam filter are described below.

2.2.1 Single Classifier models:

E -mail filtering is done by either knowledge engineering, machine learning or a combination of both. The machine learning approach is more efficient than knowledge engineering as it doesn't require rules to be specified explicitly. There are many algorithms for machine learning approach.

Pantel *et al* [19], first proposed the use of the Bayesian framework for text classification. Furthermore, Sahami *et al* [23] used Bayesian approach to junk mail filtration. The naive Bayes algorithm works on the principle of calculating the posterior probability of the feature in terms of prior probabilities of class and feature.

It is based on Bayesian theorem and basically uses maximum likelihood for classification into two classes. In the Bayesian model, the probability of message representation, denoted as $x = [x_1, x_2, \ldots, x_i]$ belongs to a class $c \in \{s, l\}$, is represented as,

$$P(c|x_i) = \frac{P(x_i|c)P(c)}{P(x_i)} \quad (1)$$

Where, local probability distribution, $P(x_i|c)$, for each attribute is specified by conditional probability table, and $P(c)$ is the probability that a message is classified as class c. For example, in the figure 8, by calculating the prior probability and likelihood of the white object in the domain.

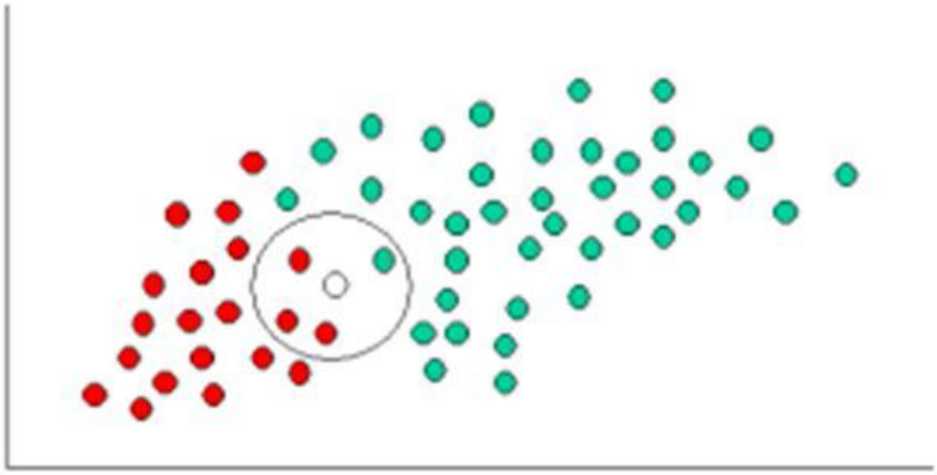


Figure 8 White object is classified to the GREEN or RED

Drucker *et al* [6] proposed the use of Support Vector machines and compared the technique with two different approaches, boosted decision tree (using C4.5 algorithm) and TF-IDF [10]. The basic concept behind Support Vector Machine is the calculation of optimal hyper plane as decision boundary which separated different classes as shown in figure 9 [22]. This is done through calculating the following optimization problem.

Maximize:

$$\sum_{i=1}^{n} \alpha_i - 1/2 \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (2)$$

Subject to:

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad (3)$$

Where $0 \leq \alpha_i \leq \mathrm{b}, i = 1, 2, \ldots, \mathrm{n}$

Here, α_i is the weight of x_i and y_i is between {-1 , +1}.

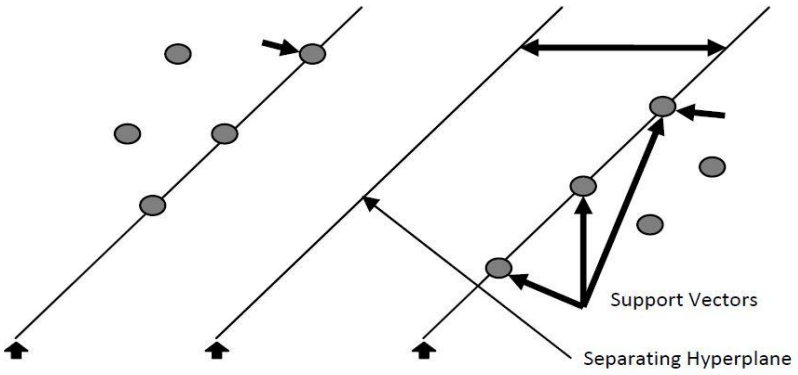


Figure 9 Separation of text using SVM

SVMs maximize the margin around the separating support vectors. Support vectors are the data points that lie closest to the decision surface.

Zdziarski *et al* [30] propose the method of KNN (K Nearest Neighbour) classifies the mail based on their similarity to the class of mails as shown in figure 10. The k – means is basically used to partition the data in such a way that while the inter cluster similarities is low, intra cluster similarities are high. KNN classifier uses Euclidean distance to find k nearest neighbours. It uses Euclidean distance for finding the k-nearest neighbours:

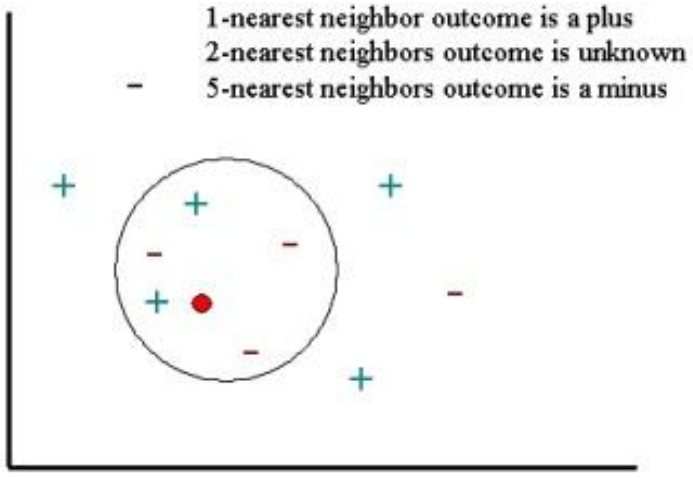


Figure 10 K-NN classifier

Euclidean distance for calculating distance between 2 points (x_i, y_i) to (x_j, y_j) is given as:

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (4)$$

For using K-NN classifier in text classification, instead of using Euclidean distance, Hamming distance is used. For example, distance between the words "words" and "world" is 1.

Kufandirimbwa *et al* [11] proposed the classification based on Artificial Neural networks (which consist of perceptron learning rule). The basic working is the creation of the word list w, which consist of n most frequently used words in the content of the labelled dataset. Artificial Neural Network input vector consist of elements such that for element i in the n^{th} position in word list, which consist of m_n instances and the total no. of words are l, entry value for l_n is set as m_n/l. Thus in this technique, input vector is created by using both frequency of word and the total number of words. After calculating the input vector, artificial neural network is employed (3 layer architecture i.e. input layer, hidden layer, output layer).

Shrivastava *et al* [25] proposed a genetic algorithm for spam filtering. It consist of multiple steps. The initial population of the Genetic Algorithm is generated randomly. The authors propose new techniques for high quality population. Reproduction consist of two different techniques, Generational Reproduction and Steady state Reproduction. Either two older generation can be couple together to form two children's, or either crossover or mutation can be applied on two individual chromosomes create ting one or two children's. Parent selection mechanism is applied on newly created chromosomes. It can be Fitness-based, Rank-based or Tournament - based. After this process, crossover, mutation and inversion is applied.

2.2.2 Multiple Classifier Models

An efficient spam filtering model requires some amount of labelled training data, either it can be supplied initially, or it can be learned from user feedback. Nearly all of the current techniques are based on using single classifiers. Apart from using a single classifier, there have been many proposals of combining different classifiers for increasing accuracy in classification problem.

Matos *et al* [16] proposed the use of global and local classifier using Bayesian networks. This model was used to classify handwritten digits. In this paper, feature space is split and then associated with the local classifiers. After this process, a Bayesian network is used to combine the outputs of the local classifiers.

Another method to achieve multi-task reinforcement learning in Bayesian process was introduced by Lazaric *et al* [12]. They proposed a hierarchal Bayesian model for (i) joint learning of observations and knowledge gain, and (ii) transfer of knowledge to assist new observed task.

Furthermore, the use of local and global classifier is supported by Zhong *et al* [31]. Instead of using the loss function as the difference between actual and calculated probabilities, loss function is calculated as $L(x_i) = \frac{1}{P^*(c|a)}$. They proposed maximizing the classification accuracy of the Naive Bayes classifier, instead of improving the maximum likelihoodness.

Another method for achieving multi-task learning is the use of hashing in the collaborative spam filtration technique [13] .Using feedback as a source of labelled data, high classification accuracy can be achieved. But this model has a drawback as users can be lazy in labelling data. So trust weights are assigned to each user.

Jatana *et al* [9] proposed the use of improved data structure for Bayesian model. Training a spam filter through tokenization is a slow process as large scale data is generated in the training process. So selection of right data structure is essential for increasing the performance of the model. Instead of storing tokens as a whole, radix encoded fragmented database approach improves the overall efficiency of the process.

Chapter 3

PROBLEM STATEMENT

In this particular chapter, the gaps which exist in the current work, problem statement of our proposed work, the objectives which are to be achieved and the method for achieving these objectives are discussed.

3.1 Gap Analysis:

In the literature survey chapter, different steps involved in Spam classification process have been discussed. Also, for each step, different techniques which can be used to create appropriate model. In the existing work following gaps exist:

1. Most of the existing techniques of Spam Classifications are based on using single classifier.
2. Due to use of single classifier, inherent disadvantages of these classifiers affects the overall accuracy of the model.
3. Different classifiers can be combined, in serial (to improve the acceleration of the model), or parallel (to improve the accuracy).
4. Traditional Bayes classifier doesn't use current knowledge to calculate the posterior probability of the tokens.
5. Instead of learning a single classifier for all users, multiple local classifiers in tandem with a global classifier can be learned.

3.2 Problem Statement

Various Spam Filter models have already been introduced for classification of mail based on certain features extracted from the message. There have been many improvements in the Naive Bayes classifier such as using radix encoded dataset instead of hashmap. Or applying different feature selection methods, pre-processing algorithms, to improve the accuracy of the classifier. But in every case, the main focus of the Naive Bayes model is to improve the maximum likelihoodness of the classifiers. As this model is based on frequency count, where on every occurrence of the token, either in good or spam mail, its frequency is increased by one. In order to overcome this problems, proposed model use local classifiers for each user and single

global classifier.

3.3 Objectives

1. To study different techniques for each step of the Naive Bayes classifier.
2. To focus on improving classification accuracy of the model instead on maximizing the likelihoodness.
3. To use current knowledge of the local classifiers for calculating the increment done for each token, instead of increasing the value by one.
4. To test and validate the model by using different spam corpuses, and comparing the result with the existing model.

Chapter 4

METHODOLOGY

This section will discuss the proposed model of spam filtration technique based on local-global classifier. This section is divided into two subparts, 1. Training the classifiers using labelled data 2. Learning incoming message according to the trained data.

4.1 Training model

Training model consist of 2 subparts, 1. Importing training data and pre-processing of labelled data 2. Tokenization of the data, calculating attributes associated with each token and training of each local and global classifier.

4.1.1 Pre-processing:

Our dataset consist of more than 36000 mail corpus. Each email consists of additional features such as sender, receiver information, timestamp of email, images etc. For our proposed model only text content is required. So in this process, all the training data are checked against pre-processing model to remove additional information and multimedia data.

4.1.1.1 Porter Stemming Algorithm

First step in our model is to run stemming algorithm reduce the number of tokens in the end process. Derivationally related words are reduced to root words to reduce the feature space and runtime of the overall algorithm. For proposed model, porter stemming algorithm is used.

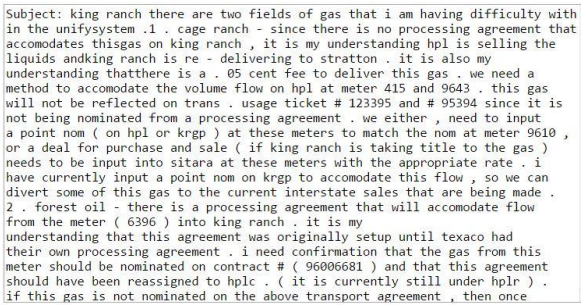

Subject: king ranch there are two fields of gas that i am having difficulty with
in the unifysystem .1 . cage ranch - since there is no processing agreement that
accomodates thisgas on king ranch , it is my understanding hpl is selling the
liquids andking ranch is re - delivering to stratton . it is also my
understanding thatthere is a . 05 cent fee to deliver this gas . we need a
method to accomodate the volume flow on hpl at meter 415 and 9643 . this gas
will not be reflected on trans . usage ticket # 123395 and # 95394 since it is
not being nominated from a processing agreement . we either , need to input
a point nom (on hpl or krgp) at these meters to match the nom at meter 9610 ,
or a deal for purchase and sale (if king ranch is taking title to the gas)
needs to be input into sitara at these meters with the appropriate rate . i
have currently input a point nom on krgp to accomodate this flow , so we can
divert some of this gas to the current interstate sales that are being made .
2 . forest oil - there is a processing agreement that will accomodate flow
from the meter (6396) into king ranch . it is my
understanding that this agreement was originally setup until texaco had
their own processing agreement . i need confirmation that the gas from this
meter should be nominated on contract # (96006681) and that this agreement
should have been reassigned to hplc . (it is currently still under hplr) .
if this gas is not nominated on the above transport agreement , then once

Figure 11 Original E-mail Content

The Porter Stemming module receives a word and returns a new token in its original form. It works by removing the suffix from word and checking that word against root form.

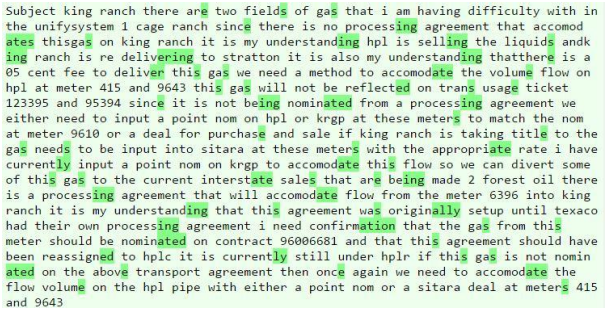


Figure 12 Highlighted characters show overlay

By returning the words to their root form, the training data size can be greatly reduced, as well as increasing the probability of the word.

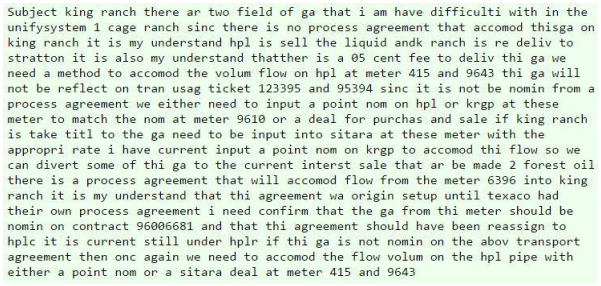


Figure 13 Stemmed Email by removing overlay

This results in the overall accuracy improvement of the classifier. The algorithm is applied by the system when any new word is presented. The training data only store the processed tokens returned by the algorithm.

4.1.1.2 Removing Stop- words and smoothing data:

There are many 'stop words' in English. For example, 'A', 'an', 'is' etc. The frequency of these words is generally very high, but they don't convey any message with respect to the content of the message. And they occupancy significant amount of space when storing in the data structure. This results in the overhead in computation and increase in error in the accuracy of the model. Also, there are some words, namely rare words, such that their frequency is very low. These words don't have a significant impact in the classification process. Thus removal of all those ' stop words ' and rare words from our data set in the preprocessing step is done.

4.1.2 Tokenization and training:

This is the learning step, which teaches the filter which e-mail is spam and what is genuine. After pre-processing, all the stop words are removed and the content is summarized. Additional information is removed from all the mails. The model is

provided with two sets of labeled data into the system. For each e-mail, the message content is broken down into smaller words. A word is a nothing but a consecutive sequence of characters. Two words may be separated by one or many spaces. After tokenization, attribute assignment for all the tokens is performed. In this step each token is provided with some attributes. These attributes are:

- GoodCount

  In traditional Naive Bayes classifier, this attribute represents the frequency of a token in good mail. Every time, when a token is found in the good mail (ham), frequency count is increased by one. In our propose model, instead of increasing the count by one, the increment is done with respect to the difference in the posterior probability difference of the token, computed using all the local classifiers and global classifier.

- BadCount:

  It is similar to the Good Count attribute. Instead of looking token in the good mail, detection in spam is performed. The increment is similar to proposed good count increment.

- GoodProb:

  This attribute defines the probability of a token to be present in the good mails. Once all the mails are tokenized and goodcount is assigned to them in the probability table, for each token Good Probability is calculated using the following equation:

$$P(token_i) = \frac{GoodCount(token_i)}{\sum_{i=1}^{n} GoodCount(token_i)} \quad (5)$$

- BadProb:

  This attribute defines the probability of a token to be present in the spam mails. Once all the mails are tokenized and BadCount is assigned to them in the probability table, for each token Bad Probability is calculated using the following equation:

$$P(token_i) = \frac{BadCount(token_i)}{\sum_{i=1}^{n} BadCount(token_i)} \quad (6)$$

- SpamProb:

  As the name suggests, this attribute defines the spam probability of a token using good probability and bad probability respectfully. It is calculated using the following equation:

$$SpamProb(token_i) = \frac{BadProb(token)}{BadProb(token)+GoodProb(token)} \quad (7)$$

The training data is encoded into a probability table. This table stores the probability of every word found in the training data, categorized into Spam and Non-spam classes. Thus, for some training data, a particular word will have a different probability for the Spam class and a different probability for the Non-spam class. In general, Naive Bayes classifier is based on frequency counting for every tokens. This approach result in computing the maximum likelihood parameters (i.e. maximizing the likelihood of the labeled training data). In this approach, while learning the CPT for a feature n_i in class c_i , i.e. $F\ (c, xi)$, frequency count is increased by one. This method of parameter learning is known as Frequency Estimate. Following is the explanation of the traditional Naive Bayes Classifier and our proposed model.

Algorithm 1 Traditional Naive Bayes algorithm for training classifier

1: Initialize each CPT entry as 0

2: **for** each message from 1 to n **do**

3: **for** each token from 1 to m

4: Increase frequency count of x_i in class c by 1

5: **end for**

6: **end for**

7. $P(x_i|c) \leftarrow \frac{F(c,x_i)}{\sum_{x_i} F(c,x_i)}$

In proposed model, instead of increasing frequency count by one, increment is done through posterior probability calculation across all users for that feature.

Formula for calculating posterior probability is given below:

$$P^*(c \mid x) = \alpha P(c) \prod_{i=1}^{n} P(x_i \mid c) \quad (8)$$

Where, α is known as normalization factor, $P(x_i|c)$ is the local probability density, usually represented by CPT. $P(C = c)$ is the probability of class variable, i.e. prior probability. And $P(c \mid x)$ is the probability density of the feature learnt through each local classifier. It basically guides the amount of contribution added to the CPT, thus reflecting the confidence on current NB classifier.

Each conditional probability $P(x_i \mid c)$ in a CPT is calculated using the frequency count obtained from training data as:

$$P(x_i|c) = \frac{Frequency(x_i)}{\sum Frequency(x_i)} \quad (9)$$

Algorithm 2. Proposed Algorithm based on local-global classifier

---

1: Initialize each CPT entry as 0

2: **for** each message from 1 to n do

3: **for** each tokens 1 to m **do**

4: Calculate posterior probability $P^*(c \mid x_i)$ using current local classifiers, using equation (8)

5: Use eq. (10) for calculating $L(x_i)$.

6: Increase $F\ (c, x_i)$ by, $\Delta F\ (c, x_i)\ =\ L(xi)$

7: **end for**

8: **end for**

Local-global classifier is a discriminative frequency estimate for Bayesian network classifier [4, 26]. In traditional classifiers, increment is based on predefined accuracy count, thus posterior probability is of no use when training for new incoming message [17]. In doing so, current knowledge of the local classifiers is not used. But in our approach, for any instance, the current classifier is used to update the CPT entries instead of simple increment.

In general, difference between the predicted probability $P^*(c \mid x_i)$ and true probability $P(c \mid x_i)$ is the value of increment in the corresponding entries. Loss function is defined as $L(x_i)$, for a feature x_i as:

$$L(x_i) = P(c|x_i) - P^*(c|x_i) \quad (10)$$

As the data used is labelled data $P(c|x_i) = 1$. The basic idea is that if a feature is already correctly classified, then the amount of increment is minimum. But if training example is misclassified, then large value is added to the CPT entry to strengthen the belief.

4.2 Classifying a new email

Learning model used is similar to traditional Naive Bayes Classifier, with each incoming message being learnt by individual classifier. In the proposed model, knowledge is shared for those tokens which are not originally present is their respective local classifier. This method creates transfer of knowledge, and helps removing the problem of cold start for every user. First few steps of the classification process are similar to the learning phase of our model. Incoming mail is divided into tokens. After tokenization of the mails, porter stemming algorithm is applied. This reduces the incoming tokens to their root forms. It is mandatory to apply Porter Stemming algorithm in this step because if the original word is passed to the classifier, it won't be able to detect word in the probability table as in the learning phase, each token was stripped to its base word. So each word is passed through Stemming module. The overall impact of the stop words and rare words is minimum as their presence is already minimum. So these words don't affect the results at all, because the system returns an extremely small probability for any frequent words appearing as they are not found in the training data. Bayesian Filter works on the principle of maximum likelihoodness.

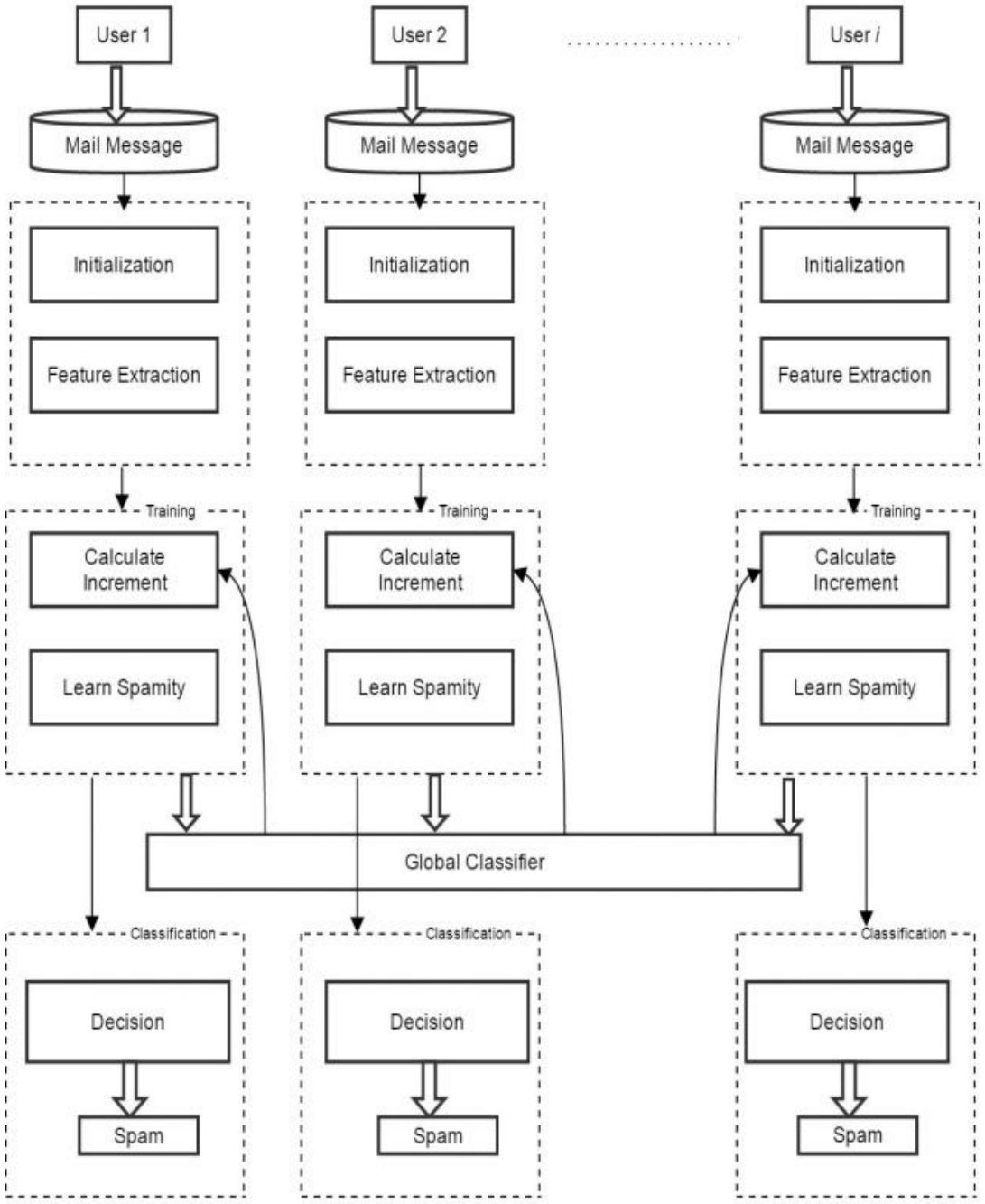


Figure 14 Local-Global classifier for spam filtration

Every incoming mail can be represented in the form of vector $v(x_1, x_2, \ldots\ldots, x_m)$, where m is the number of tokens present in the mail. So the probability of a mail to be spam or non-spam, in the Bayesian model is given as:

$$P(Spam|x_1, x_2, \ldots, x_m) = \frac{P((x_1,x_2,\ldots,x_m)|Spam)\ P(Spam)}{P((x_1,x_2,\ldots,x_m))} \qquad (11)$$

Similarly,

$$P(nonSpam|x_1, x_2, \ldots, x_m) = \frac{P((x_1,x_2,\ldots,x_m)|nonSpam)P(nonSpam)}{P((x_1,x_2,\ldots,x_m))} \qquad (12)$$

Where,

$P(spam|x_1, x_2, \dots, x_m)$ represents the probability of a mail with vector $(x_1, x_2, \dots, x_m)$ to be spam mail. Similarly $P(nonspam|x_1, x_2, \dots, x_m)$ represents Non Spam mail Probability. Following are the steps involved in the classification of incoming message as spam or legitimate message.

Step 1: Tokenization of incoming email and apply porter Stemming Algorithm.

Step 2: Search for each token in the Probability Table data structure. This step is implemented by using hashmap in Java framework. So for each token, six attributes are returned by the hashmap. If the word is present in the local classifier hashmap, then take all the attribute for further calculations. But if the word in not present in the local classifier then instead of assigning a constant predefined spam probability (Like in the traditional Naive Bayes classifier), information is gathered from the other classifier regarding the tokens spamity. This result in information sharing between different local classifiers using global classifier.

$$pSpamSet(token_i) = \begin{cases} pSpam(token_i) \mid token_i \in hashtable \\ \frac{\sum_{k=1}^{m} pSpam(token_i)}{m} \end{cases} \quad (13)$$

Step 3: Since Naive Bayes classifier works on the principle of improving the maximum likelihoodness. For each token, interestingness factor is calculated as follows:

$$Interestingness(token_i) = mod(0.5 - spamProb(token_i)) \quad (14)$$

After calculating Interestingness factor for all the tokens, hashmap for top k tokens is created. Entries in the hashmap are in the Decreasing order of Interestingness value.

Step 4: After calculating the Hashmap containing top k interestingness value tokens, positive and negative probability of the email is calculated as.

$$pposproduct(mail) = \prod_{i=1}^{k} Spamprob(token_i) \quad (15)$$

$$pnegproduct(mail) = (1 - pposproduct(mail)) \quad (16)$$

And the final probability is calculated as:

$$pspam(mail) = \frac{pposproduct(mail)}{pposproduct(mail) + pnegproduct(mail)} \quad (16)$$

In proposed model, the threshold value for a mail to be considered as spam is 0.9. So if pspam is less than 0.9 then mail is considered as legitimate mail else spam.

Chapter 5

IMPLEMENTATION & RESULTS

This project is implemented on 64-bit windows 8.1 system having 4 GB of RAM. All the implementation work is done under Java framework. Radix encoded fragmented data structure is more efficient in place of hashmap data structure. This reduces searching time for the token in the database, which leads to an overall increase in the performance of the model.

5.1. Data used

In proposed model, Enron spam dataset is used for testing. Enron spam dataset consisting of six different datasets namely enron1, enron2... enron6. Each consisting of nearly 5500 messages, labelled as spam or ham. For this experiment, it has been established that the number of users are 3. A random division of dataset for training and testing purpose ensures accurate prediction. For example, division can be 70% data as training set and the remaining data as testing. Other ratios are included in the overall computation. Furthermore, each of training and testing data is equally divided among the users. In order to remove bias for a particular user, random allocation of data to every user is done. This process is repeated for 10 times during which accuracy of the proposed model is measured in terms of deviation in the results.

First, the raw data is parsed for removing unwanted information and stopwords from the dataset. Then tokenization process is applied during which each token is assigned some attributes. This attributes control the spamity of a token, and are calculated using classifiers. With the introduction of each token, posterior probability of that token is calculated from all the local classifiers.

For calculation the difference between Naïve Bayes and proposed model, following evaluation measures are used. In the Table 5.1, $no_{h \to h}$ represents the number of Ham messages classified as Ham. Similarly $no_{h \to s}$ represents the Ham messages classified as Spam.

Table 1 Different Evaluation Measures

Evaluation Measure	Evaluation Function
Accuracy	$\frac{no_{h\to h} + no_{s\to s}}{no_{h\to h} + no_{s\to s} + no_{h\to s} + no_{s\to h}}$
Error Rate	$\frac{no_{h\to s} + no_{s\to h}}{no_{h\to h} + no_{s\to s} + no_{h\to s} + no_{s\to h}}$
False Positive Rate	$\frac{no_{h\to s}}{no_{h\to h} + no_{h\to s}}$
False Negative Rate	$\frac{no_{s\to h}}{no_{s\to s} + no_{s\to h}}$
Recall	$\frac{no_{s\to s}}{no_{s\to s} + no_{s\to h}}$
Precision	$\frac{no_{s\to s}}{no_{s\to s} + no_{h\to s}}$

5.2. Implementation Result

Experimental results have observed high accuracy rate with small deviation in classification of messages. It is observed that nearly 95% of the ham messages are correctly classified for each user. Also in case of spam message, nearly 93% of the labelled data is correctly classified. As the number of labelled data increases, classification accuracy of the model improves. Following chart contains accuracy of our proposed hybrid classifier for all the 6 Enron datasets.

Here, user1_g represents the legitimate mail detection accuracy of 1st user, similarly user1_s represents the spam detection accuracy. Similarly for other 2 users. Also, horizontal axis represent the percentage of data used as training data. As evident from the fig. Legitimate mail detection accuracy saturates around 45 % usage of data as training data. All the results shown are derived through running project 10 times, and calculating the average and deviation in the results. Experimental results show significantly small amount of deviation in the results. Figure 5.1, shows the user accuracy for both spam and ham messages, while using Enron 1 dataset.

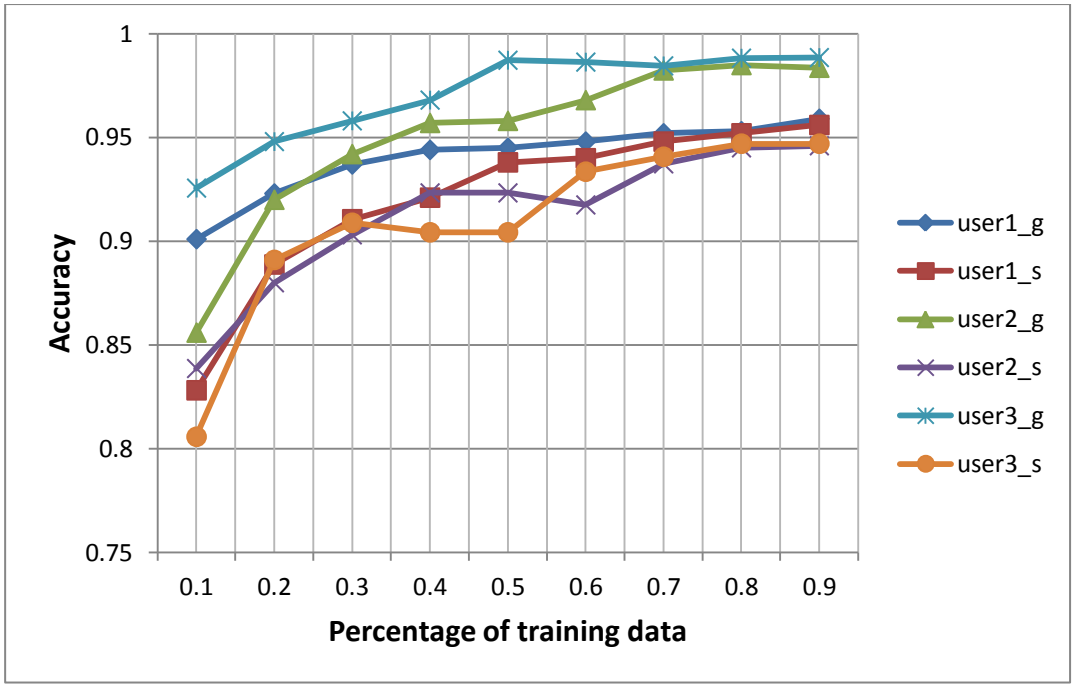


Figure 15 Enron 1 Data Set Accuracy in spam detection for all users

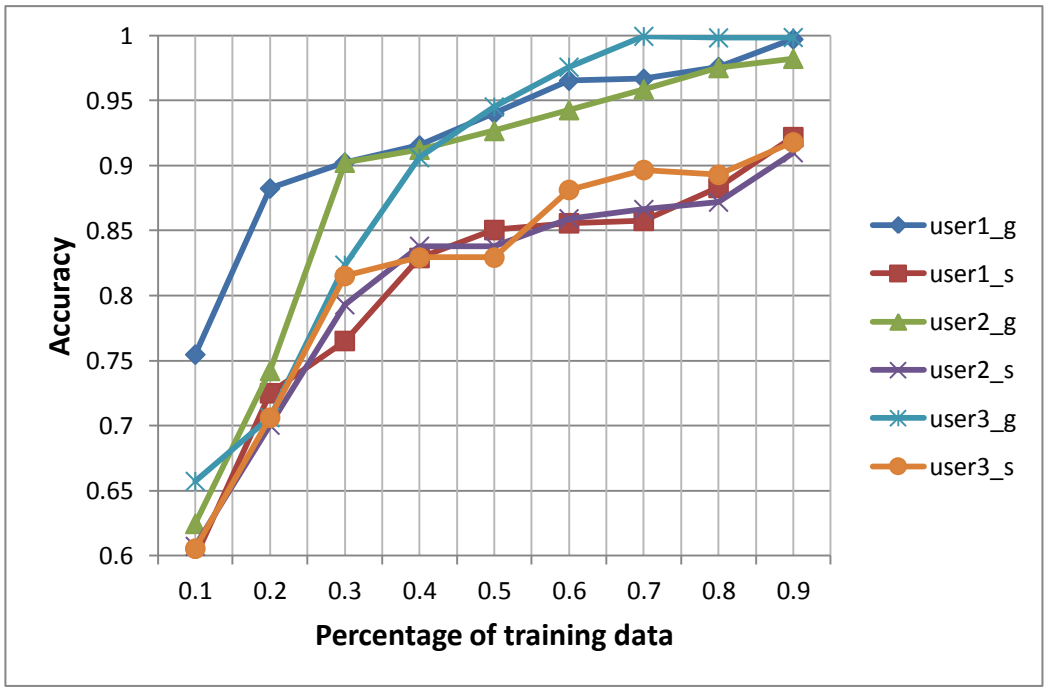


Figure 16 Enron 2 Data Set Accuracy

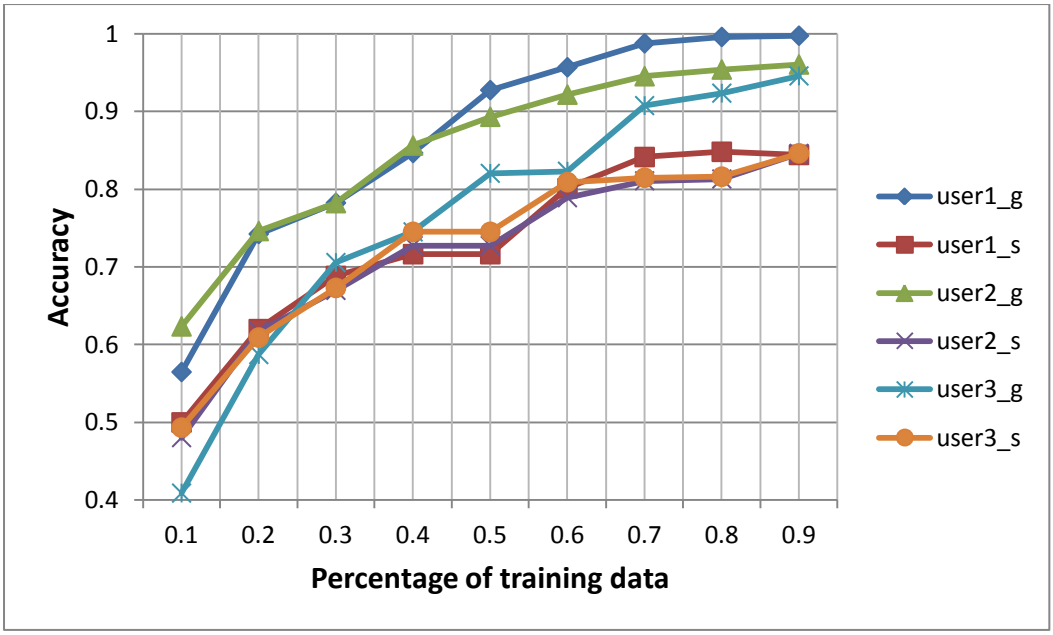


Figure 17 Enron 3 Data Set Accuracy

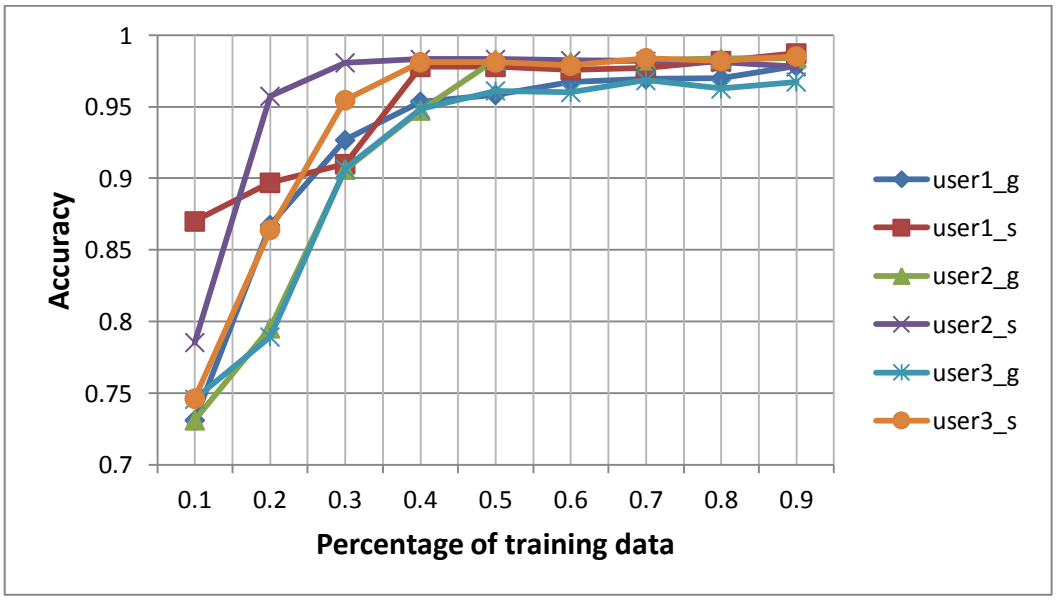


Figure 18 Enron 4 Data Set Accuracy

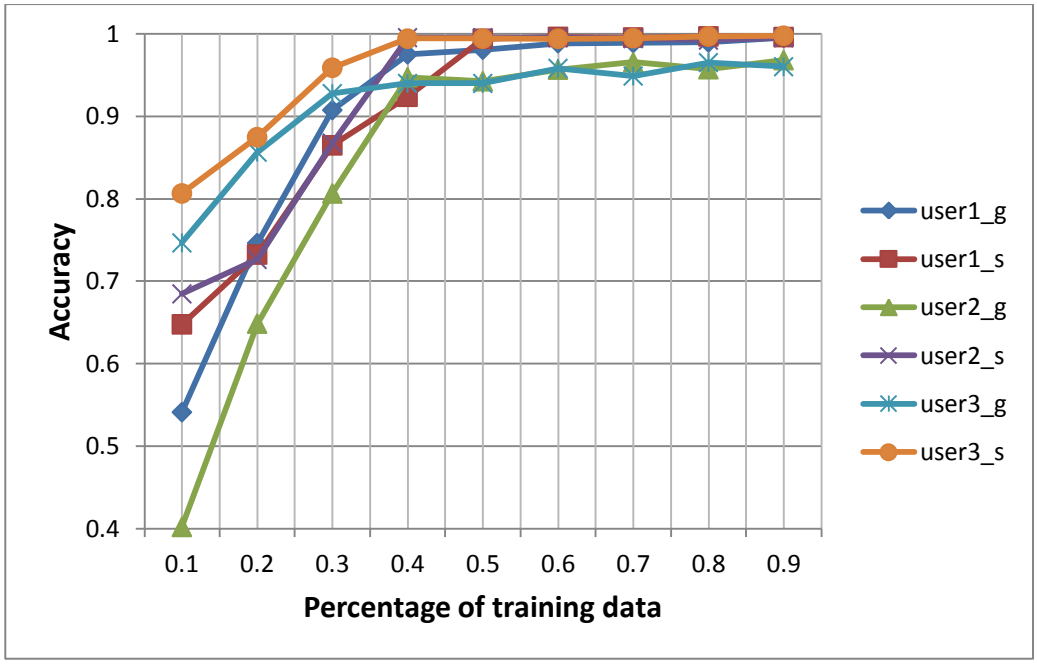


Figure 19 Enron 5 Data Set Accuracy

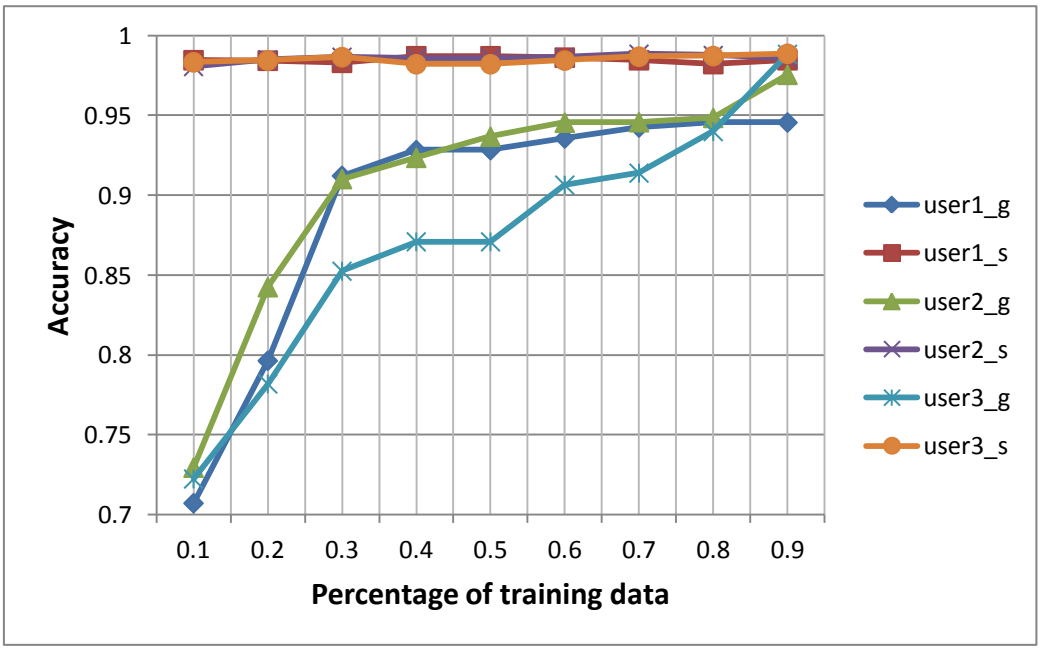


Figure 20 Enron 6 Data Set Accuracy

The saturation point is achieved at 45% of training data. As it can be observed that the overall accuracy of the spam and legitimate classification is similar for all the users, comparison is done with the Naive Bayes classifier using single user. Table 2.1 is used to calculated parameters of proposed model, and the results are compared with

the traditional Naive Bayes algorithm. Our experimental results show significant improvement in terms of Error rate, Spam Recall rate and Spam Precision. For some part of the dataset, original classifier works better than our proposed model, but for most of the part results are reversed.

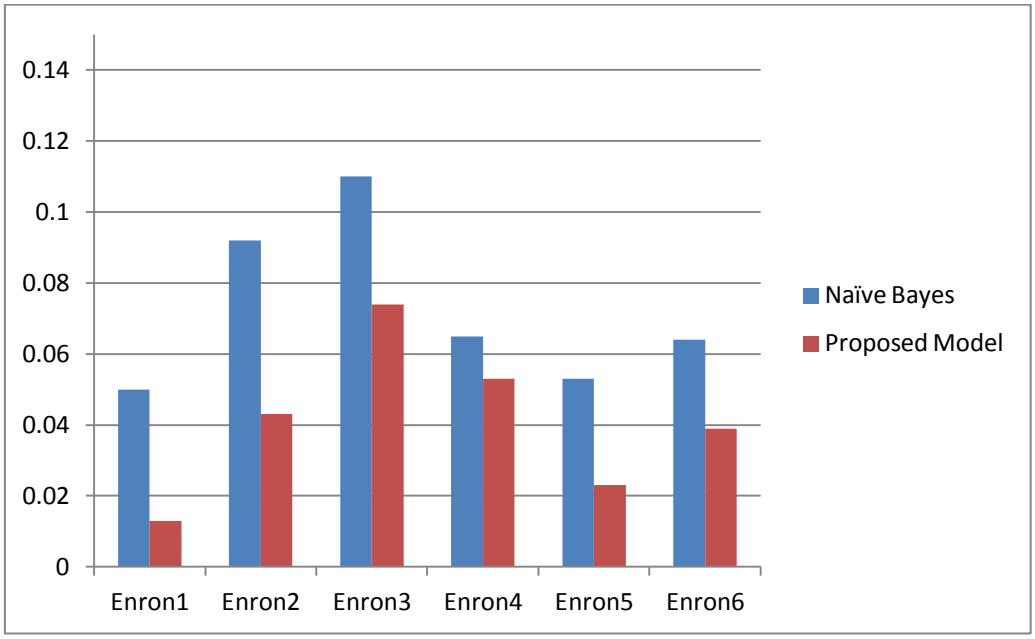


Figure 21 Error rate comparison

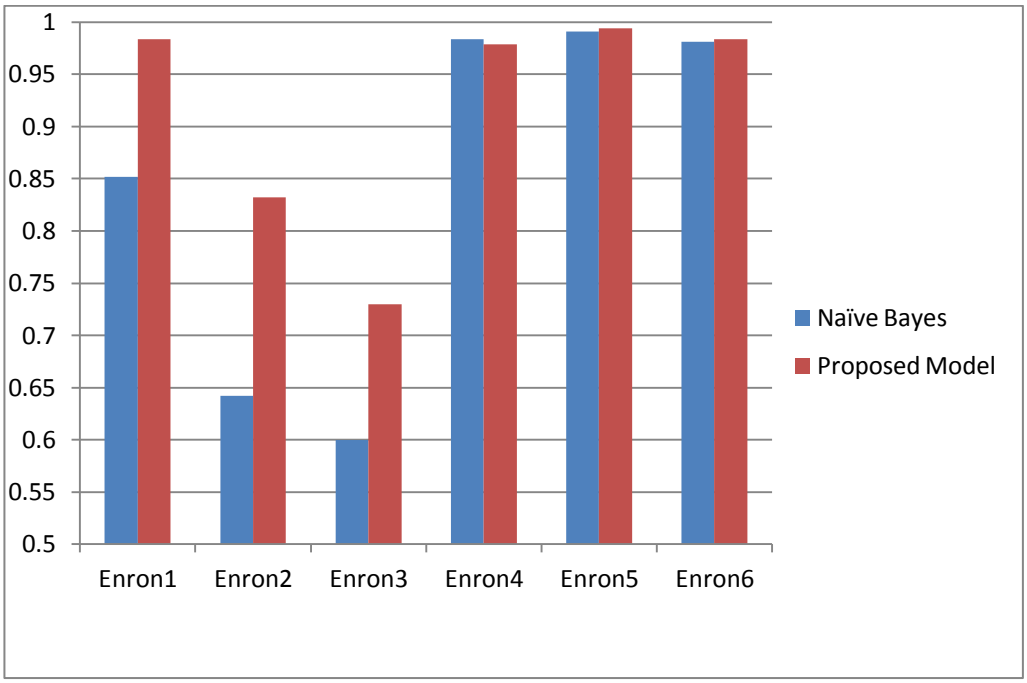


Figure 22 Spam Recall rate comparison

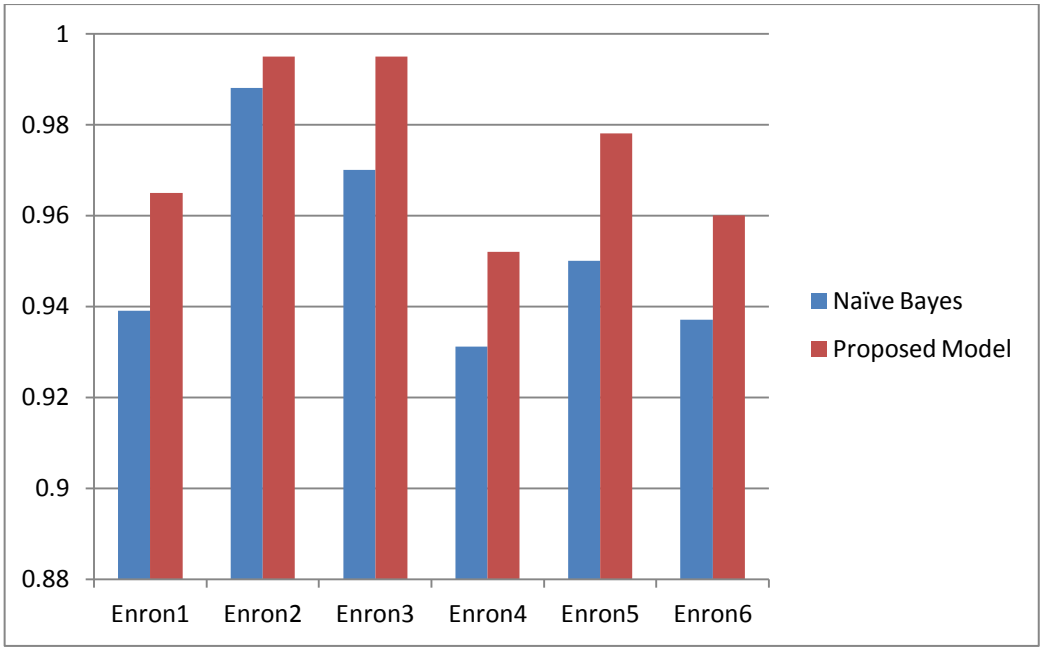


Figure 23 Spam Precision rate comparison

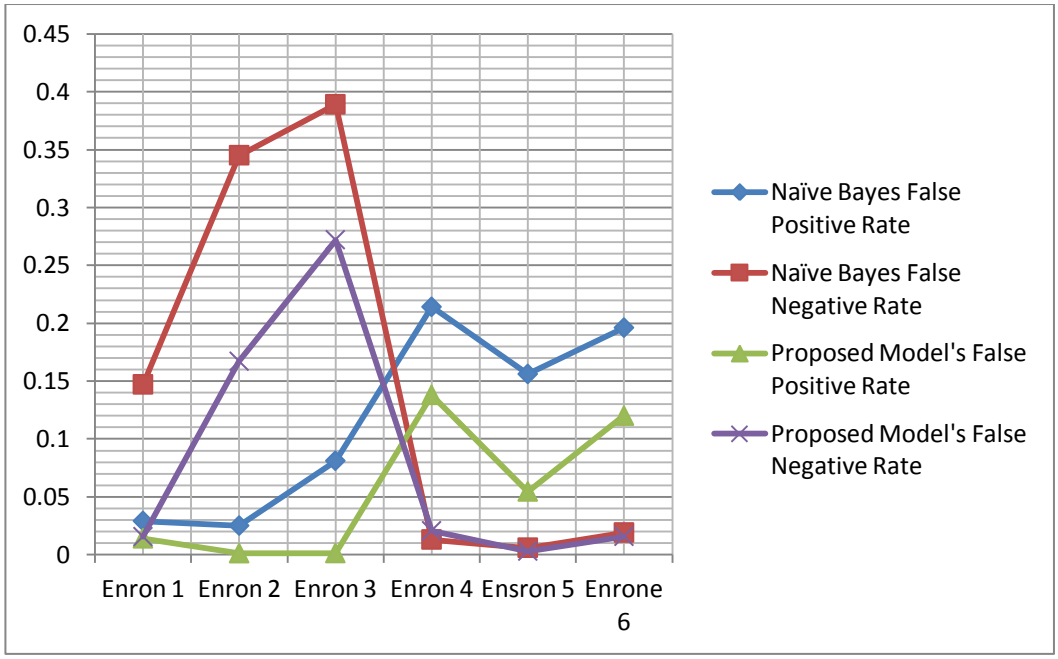


Figure 24 False Positive and False Negative Comparison

Table 2 Test results of proposed model (60% training set)

		User 1	User 2	User 3
Enron1	Ham	**0.986**±0.007	**0.975**±0.0014	**0.989**±0.002
	Spam	**0.967**±0.0024	**0.972**±0.0068	**0.954**±0.008
Enron2	Ham	**0.914**±0.0074	**0.943**±0.0016	**0.923**±0.001
	Spam	**0.897**±0.089	**0.884**±0.0076	**0.951**±0.0049
Enron3	Ham	**0.927**±0.007	**0.976**±0.0041	**0.964**±0.0016
	Spam	**0.981**±0.047	**0.988**±0.0066	**0.988**±0.0044
Enron4	Ham	**0.882**±0.0058	**0.874**±0.0075	**0.862**±0.0010
	Spam	**0.974**±0.0002	**0.963**±0.0016	**0.996**±0.0240
Enron5	Ham	**0.889**±0.0076	**0.971**±0.0057	**0.956**±0.0079
	Spam	**0.921**±0.083	**0.937**±0.0043	**0.947**±0.0046
Enron6	Ham	**0.972**±0.0076	**0.982**±0.0017	**0.927**±0.0014
	Spam	**0.947**±0.0019	**0.890**±0.049	**0.852**±0.0074

Chapter 6

Conclusion and Future Scope

6.1 Conclusion:

A Hybrid Bayesian classifier is presented, with local classifier for each users, and a global classifier which governs the parameters of tokens in the local classifier. This leads to a type of information sharing among the users while maintaining the individuality. Experimental results show that this model improves the overall accuracy of classification among users. Naive Bayes is a simple and efficient technique of spam filtration. Creating a Naive Bayes network helps in exploiting the commonness among different tasks, thus learning and modifying accordingly. Following objectives are achieved in this dissertation:

- Improved classification accuracy of Naive Bayes spam filtering algorithm.
- More emphasis on classification accuracy rather than maximizing likelihoodness.
- Sharing of knowledge among different users and use of current knowledge for improving performance.

6.2 Future Scope:

There are many models of spam classification and filtration. Many of them are based on using single classifier. But in recent times, the use of multiple classifiers is advocated and research is done by using a combination of statistical techniques during different phase of the spam filtration. In future, research can be conducted to further improving the classification accuracy.

References

[1] Almeida, T. A., Almeida, J., & Yamakami, A. (2011). Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers.Journal of Internet Services and Applications, 1(3), 183-200.

[2] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D. (2000, July). An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 160-167). ACM.

[3] Awad, W. A., & ELseuofi, S. M. (2011). Machine Learning methods for E-mail Classification. International Journal of Computer Applications (0975–8887), 16(1).

[4] Bouchard, G., & Triggs, B. (2004). The tradeoff between generative and discriminative classifiers. In 16th IASC International Symposium on Computational Statistics (COMPSTAT'04) (pp. 721-728).

[5] Cormack, G. V., Smucker, M. D., & Clarke, C. L. (2011). Efficient and effective spam filtering and re-ranking for large web datasets. Information retrieval, 14(5), 441-465.

[6] Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. Neural Networks, IEEE Transactions on, 10(5), 1048-1054.

[7] Frakes, W. B., & Fox, C. J. (2003, April). Strength and similarity of affix removal stemming algorithms. In ACM SIGIR Forum (Vol. 37, No. 1, pp. 26-30). ACM.

[8] Garcia, F. D., Hoepman, J. H., & Van Nieuwenhuizen, J. (2004). Spam filter analysis. In Security and Protection in Information Processing Systems (pp. 395-410). Springer US.

[9] Jatana, N., & Sharma, K. (2014, March). Bayesian spam classification: Time efficient radix encoded fragmented database approach. In Computing for Sustainable Global Development (INDIACom), 2014 International Conference on (pp. 939-942). IEEE.

[10] Joachims, T. (1996). A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization (No. CMU-CS-96-118). Carnegie-mellon univ pittsburgh pa dept of computer science.

[11] Kufandirimbwa, O., & Gotora, R. (2012). Spam Detection using Artificial Neural Networks (Perceptron Learning Rule). Online J Phys Environ Sci Res, 1(2), 22-29.

[12] Lazaric, A., & Ghavamzadeh, M. (2010, June). Bayesian multi-task reinforcement learning. In ICML-27th International Conference on Machine Learning (pp. 599-606). Omnipress.

[13] Li, G., Hoi, S. C., Chang, K., Liu, W., & Jain, R. (2014). Collaborative Online Multitask Learning. Knowledge and Data Engineering, IEEE Transactions on, 26(8), 1866-1876.

[14] Li K., Zhong Z.: Fast statistical spam filter by approximate classifications . In Proceedings of the Joint international Conference on Measurement and Modeling of Computer Systems. Saint Malo, France, 2006

[15] Malarvizhi, R., & Saraswathi, K. (2013). Content-Based Spam Filtering and Detection Algorithms-An Efficient Analysis & Comparison. International Journal of Engineering Trends and Technology (IJETT), 4(9).

[16] Matos, L. N., & De Carvalho, J. M. (2006, August). Combining global and local classifiers with Bayesian network. In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on (Vol. 4, pp. 952-952). IEEE.

[17] McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, pp. 41-48).

[18] Michelakis, E., Androutsopoulos, I., Paliouras, G., Sakkis, G., & Stamatopoulos, P. (2004). Filtron: A learning-based anti-spam filter. In proceedings of the 1st conference on email and anti-spam. Mountain.

[19] Pantel, P., & Lin, D. (1998, July). Spamcop: A spam classification & organization program. In Proceedings of AAAI-98 Workshop on Learning for Text Categorization (pp. 95-98).

[20] Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130-137.

[21] Porter, M. F. (2001). Snowball: A language for stemming algorithms.

[22] Rakse, S. K., & Shukla, S. (1819). Spam classification using new kernel function in support vector machine. International Journal on Computer Science and Engineering, 2(5), 2010.

[23] Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998, July). A Bayesian approach to filtering junk e-mail. In Learning for Text Categorization: Papers from the 1998 workshop (Vol. 62, pp. 98-105).

[24] Sharma, A., & Rastogi, V. (2014). Spam Filtering using K mean Clustering with Local Feature Selection Classifier. International Journal of Computer Applications, 108(10).

[25] Shrivastava, J. N., & Bindu, M. H. (2014). E-mail spam filtering using adaptive genetic algorithm. International Journal of Intelligent Systems and Applications (IJISA), 6(2), 54.

[26] Su, J., Zhang, H., Ling, C. X., & Matwin, S. (2008, July). Discriminative parameter learning for Bayesian networks. In Proceedings of the 25th international conference on Machine learning (pp. 1016-1023). ACM.

[27] W. W. Cohen, "Learning rules that classify e-mail," in Proc. 1996 AAAI Spring Symp. Inform. Access.

[28] Wilbur, W. J., & Sirotkin, K. (1992). The automatic identification of stop words. Journal of information science, 18(1), 45-55.

[29] Zdziarski, J. A. Ending Spam (2005): Bayesian Content Filtering and the Art of Statistical Language Classification. No Starch Press, San Francisco, CA, USA.

[30] Zhang, H., Berg, A. C., Maire, M., & Malik, J. (2006). SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. InComputer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on (Vol. 2, pp. 2126-2136). IEEE.

[31] Zhong, S., & Langseth, H. (2009, December). Local-global-learning of naive Bayesian classifier. In Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on (pp. 278-281). IEEE.

List of Publications & Video Link

Solanki R. K., Verma K., Kumar R. (2015). Spam Filtering Using Hybrid Local-Global Naive Bayes Classifier. Accepted in ICACCI Special Session on Advances in Data and Knowledge Engineering. Kochi, Kerala.

Video link of the project: https://www.youtube.com/watch?v=kWA4Ersk_6E